ЗП

УДК 004.415.3

***О.Н. Паулин***
Одесский национальный политехнический университет, Украина
Украина, 65044, г. Одесса, пр. Шевченко, 1, *paulin@te.net.ua*

# Особенности свёртки кодов при построении умножителей на основе метода ромбов

***O.N. Paulin***
*Odessa National Polytechnic Unversity, Ukraune, c. Odessa*
*65044, 1, Shevchenko ave., Odessa, Ukraine, paulin@te.net.ua*

# *Features of Codes Compressing by Multiplier Design on the Base of the Rhombus Method*

***О.М. Паулін***
Одеський національний політехнічний університет, Україна
Україна, 65044, м. Одеса, пр. Шевченка, 1

# Особливості згортки кодів при побудові помножувачів на основі методу ромбів

Рассматриваются варианты обработки старших $n/2$ разрядов при проектировании умножителей $n\times n$ на основе сумматоров типа ромб: использование модифицированного условного переноса и расширение в 1,5 раза разрядности трёхоперандного сумматора. Наилучшее в смысле минимума задержки решение определяется разрядностью $n$ сомножителей.
**Ключевые слова:** быстродействующий умножитель, ромб бит частичных произведений, декомпозиция, сумматор типа ромб, трёхоперандный сумматор, вырожденный сумматор, условный перенос.

The variants of processing of $n/2$ ms bits by design rapid multipliers $n$x$n$ on the base of adders of the rhombus type are considered. These are: using of improved condition transfer and extension in 1.5 once of digit length of the three-operand adder for account of inclusion ms bits in this length. Best decision is determined by digit length $n$ of the factors.
**Key Words:** rapid multipliers, rhombus of bit, decomposition, adder by rhombus type, three-operand adder, devolution adder, conditional transfer.

Розглядаються варіанти обробки старших $n/2$ розрядів при проектуванні швидкодіючих помножувачів $n\times n$ на основі суматорів типу ромб: використання модифікованого умовного переносу і розширення розрядності трьохоперандного суматора в 1,5 рази за рахунок включення до неї старших розрядів. Краще рішення визначається розрядністю $n$ співмножників.
**Ключові слова:** швидкодіючий помножувач, ромб біт часткових добутків, декомпозиція, суматор типу ромб, трьохоперандний суматор, вироджений суматор, умовний перенос

## Введение

В последние годы широкое распространение получила многооперандная обработка потока данных, в основе которой лежит свёртка многорядных бинарных кодов (МРК) с помощью многооперандных сумматоров [1], [2].

Свёртка МРК используется и при аппаратной реализации схемы «школьного» умножения двух двоичных сомножителей. Совокупность строк (частичных произведений) образует при одинаковой разрядности сомножителей ромб бит частичных произведений (БЧП), который может быть интерпретирован как МРК. Тогда произведение получается как результат свёртки ромба БЧП. Однако такая свёртка за один операционный цикл при большой разрядности сомножителей нереализуема из-за высоких затрат оборудования, поэтому на практике прибегают к декомпозиции исходного ромба БЧП. В [3] предложен метод построения быстродействующего умножителя, основанный на декомпозиции ромба БЧП на одинаковые подромбы, для аппаратной реализации которых используются специализированные сумматоры типа ромб. Однако в [3] не решён вопрос построения вырожденных сумматоров.

**Постановка задачи исследования.** Требуется доработать метод ромбов и предложить новые схемотехнические решения при построении умножителей с целью повышения их быстродействия.

# Построение быстродействующих умножителей методом ромбов

Предложенный в [3] метод включает в себя:

1) декомпозицию ромба БЧП на 4 подромба половинной размерности;

2) использование для обработки подромбов сумматоров типа ромб;

3) свёртку результатов предыдущего этапа суммирования с помощью трёхоперандного сумматора (ТОС) [4].

Этот метод иллюстрируется примером схемы умножения 16×16 (рис. 1). Как и в [3], исходный ромб БЧП разбивается на 4 одинаковых подромба двумя сечениями, горизонтальным и наклонным, параллельными границе исходного ромба. Выделены 4 подромба, которые являются фрагментами 8×8 исходного ромба БЧП; они пронумерованы от 1 до 4. Эти фрагменты могут быть далее разбиты на более мелкие подромбы размером 4×4 (показано на примере подромба 1). Значком «x» обозначены биты операндов, биты ромба БЧП, биты результатов свёртки подромбов 1..4, а также биты результата перемножения. Кроме того, выделен формат результирующего суммирования, в котором показано количество слагаемых в данном разряде. Пунктиром отмечены разряды, в которые возможен перенос.

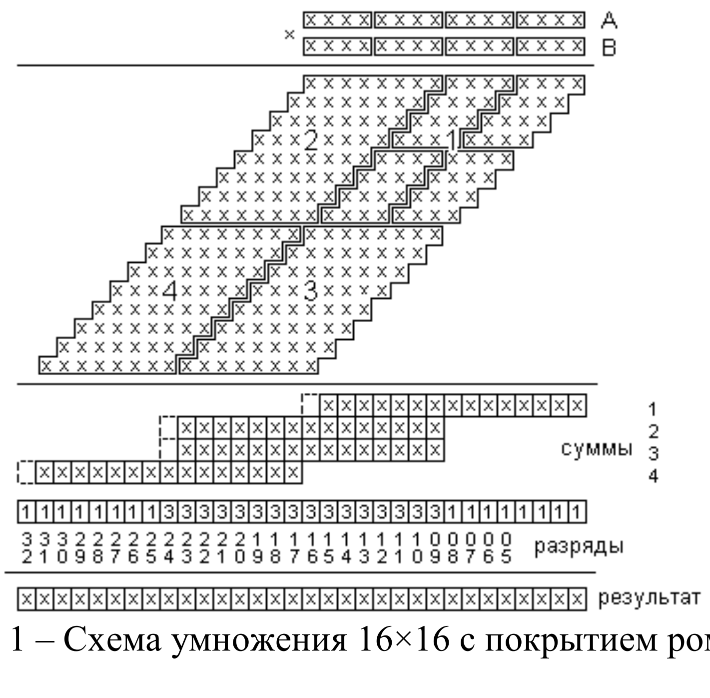



Рисунок 1 – Схема умножения 16×16 с покрытием ромбами

Каждый подромб бит обрабатывается своим сумматором типа ромб, который выдаёт однорядный результат. Для свёртки исходного ромба БЧП нужно предварительно свернуть каждый отдельный подромб, а для свёртки подромба минимального размера необходимо иметь (построить) сумматор типа ромб со следующим распределением бит слагаемых по разрядам [1 2 3 4 3 2 1].

Рассмотрение формата результирующего суммирования показывает, что первые 8 однобитовых разрядов идут непосредственно на выход в качестве готового результата, следующие 16 трёхбитовых разрядов должны быть обработаны ТОС, и старшие 8 разрядов – вновь однобитовые. Обработка последних имеет особенности, поскольку из двадцать четвёртого разряда возможен двухбитовый непозиционный перенос. Здесь возможны 2 варианта обработки старших $n/2$ разрядов: 1) использование модифицированного условного переноса для управления выбором заготовленных результатов обработки этих разрядов вырожденными сумматорами; 2) неявный учёт переноса при включении этих разрядов в общую разрядность ТОС, в результате чего она возрастает в 1,5 раза.

Из рассмотрения рис. 1 следует, что форматы результатов обработки подромбов (разряды с девятого по двадцать четвёртый) составляют трёхрядный код. Аналогичное положение будет и в общем случае, что устанавливает

**Теорема.** При покрытии ромба БЧП, полученного умножением A×B разрядностью $n \times n$, четырьмя одинаковыми подромбами результаты суммирования их бит выстраиваются в 3 ряда кодов разрядностью $n$.

*Доказательство.* Предварительно определим некоторые понятия. Назовём *длиной* формата результата его количество бит с учётом бита переноса; *сдвигом* данного формата относительно формата результата обработки 1-го ромба назовём расстояние, измеряемое количеством бит между первыми битами данного формата и формата 1-го результата.

Далее рассмотрим форматы результатов при суммировании бит ромба, включающего в себя 4 подромба размерностью $n/2 \times n/2$. Сомножители A и B выражаются через половинные форматы, то есть в результате перемножения получим A×B= =A1×B1 + A2×B1 + A2×B1 + A2×B2. Каждому слагаемому этого выражения соответствует свой ромб (1, 2, 3 и 4-й – рис. 1). Длина формата каждого результата с учётом возможного переноса из ромба равна $n$, поскольку перемножаются сомножители одинаковой разрядности $n/2$. Очевидно, сдвиги форматов результатов перемножения A1×B2 и A2×B1 одинаковы и равны $n/2$, а сдвиг формата результата перемножения A2×B2 равен $n$. Таким образом, начиная с $(n/2+1)$-й позиции по $(n+n/2)$-ю позицию (длина равна $n$) необходимо суммировать 3 слагаемых, то есть результаты обработки всех подромбов действительно образуют трёхрядный код.

**Следствие.** При переходе от данной размерности подромба к вдвое большей размерности при сборке результатов обработки новых подромбов также образуется трёхрядный код.

Теорема и следствие из неё обосновывают применение ТОС разрядности $n$ при построении умножителей на основе сумматоров типа ромб.

В общем случае процедура свёртки ромба БЧП на основе метода ромбов включает в себя несколько этапов; на каждом этапе к полученным подромбам также может быть рекурсивно применена декомпозиция на 4 одинаковых подромба половинной размерности. Размерность наименьшего подромба составляет 3×3, 4×4 или 5×5. Таким образом, размерность сомножителей должна выбираться по формуле: $n = m2^k$, где $m = 3, 4, 5$, а $k$ – глубина рекурсии при декомпозиции исходного ромба БЧП, $k = 1,2,\ldots$

На каждом этапе обработки, начиная со второго, осуществляется свёртка 3→1 с помощью ТОС.

Наименьшие подромбы обрабатываются одинаковыми сумматорами типа ромб. Для описания сумматоров разработана программа, позволяющая получить его таблицу функционирования (ТФ) по заданному распределению бит слагаемых по разрядам. Так, для сумматора, обрабатывающего ромб бит размерностью 4×4, построена ТФ [3] с распределением [1 2 3 4 3 2 1]. Аналогично могут быть построены ТФ сумматоров типа ромб для размерности 3×3 и 5×5. ТФ заполняются разрядными индексами симметрических функций (СФ) [5], причём индексы СФ вычисляются упомянутой программой.

На рис. 2 показана структура умножителя 16x16, построенного в соответствии с предложенным методом. Здесь обозначено: И – линейка из $n^2$ элементов И, реализующих конъюнкцию бит сомножителей $c_{ij}=a_ib_j$, $i$, $j$ =1..$n$; С_Р – линейка сумматоров типа ромб, в обозначении которых первая цифра – номер подромба 8×8, а вторая – номер подромба 4×4; далее – двухранговая пирамида из ТОС.

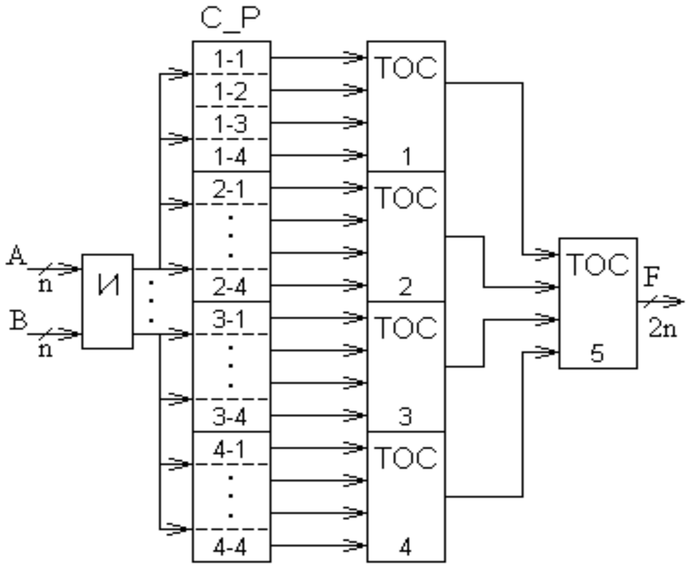



Рисунок 2 – Структура умножителя 16×16 по методу ромба

Суммирование всех бит БЧП проходит в 3 этапа: на первом этапе на сумматорах типа ромб суммируются биты «малых» ромбов, на втором этапе – на ТОС суммируются биты подромба, составленного из 4-х малых подромбов, а на последнем, третьем этапе, – на ТОС суммируются результаты предыдущего этапа. Таким образом, при предложенном подходе осуществляется свёртка 16→9→3→1.

# Обработка старших разрядов

Отметим, что при обработке старших восьми разрядов (рис. 1) возможен непозиционный перенос из ТОС 16Σ3, который может быть учтён *вырожденным сумматором* с одним операндом и несколькими битами переноса. Необходимость использования вырожденных сумматоров возникает и в общем случае.

Рассмотрим на примере умножителя 8×8 два варианта обработки старших разрядов: использование модифицированного условного переноса и расширение разрядности ТОС.

1. Описанный в [6] *условный перенос* нами модифицирован применительно к данному случаю: двухбитовый непозиционный перенос Гласера (C и C’) из ТОС 8Σ3 управляет *коммутатором* для выбора из *трёх* заготовленных результатов.

Отметим, что при сложении трёх операндов невозможен одновременный позиционный перенос в следующий разряд и через разряд, т.е. $P_1$&$P_2$=0. Это означает, что можно чётко разграничить ситуации с различными вариантами позиционных переносов: либо нет переносов (CC' = 00), либо есть перенос в следующий разряд (CC'=10), либо есть перенос через разряд (CC' = 11). Связь непозиционного переноса Гласера и позиционного переноса задаётся следующими выражениями: $P_1 = C!C'$, $P_2 = CC'$ (! означает инверсию переменной).

Коммутатор выбирает один из трёх результатов, полученных при обработке данных со следующими форматами: при отсутствия переноса из ТОС – это формат самой старшей тетрады [1 1 1 1]; в случае наличия переноса из ТОС – форматы входных данных для вырожденных сумматоров имеют вид [1 1 1 2] и [1 1 2 1] при CC' = 10 и CC' = 11 соответственно.

В первом случае результат обработки тетрады повторяет исходную тетраду – разряды с 13-го по 16-й передаются сразу на выход умножителя, во втором и третьем случаях вырожденные сумматоры одновременно подготавливают результаты до вычисления значений CC' на выходе ТОС. Описание функционирования вырожденных сумматоров приведено в табл. 1 и 2; они построены с помощью упомянутой выше программы.

Таблица 1 – РИ для сумматора [1 1 1 2], $P_1$=1

| $S_4$ | | | | $S_3$ | | | $S_2$ | | $S_1$ |
|---|---|---|---|---|---|---|---|---|---|
| 4р | 3р | 2р | 1р | 3р | 2р | 1р | 2р | 1р | 1р |
| 0 | 1 | 1 | 2 | 0 | 1 | 2 | 0 | 2 | 1 |
| 1 | 0 | X | X | 1 | 0 | X | 1 | 0,1 | |
| 1 | 1 | 0 | X | 1 | 1 | 0,1 | | | |
| 1 | 1 | 1 | 0,1 | | | | | | |

Таблица 2 – РИ для сумматора [1 1 2 1], $P_2$=1

| $S_4$ | | | | $S_3$ | | | $S_2$ | | $S_1$ |
|---|---|---|---|---|---|---|---|---|---|
| 4р | 3р | 2р | 1р | 3р | 2р | 1р | 2р | 1р | 1р |
| 0 | 1 | 2 | X | 0 | 2 | X | 1 | X | X |
| 1 | 0 | X | X | 1 | 0,1 | X | | | |
| 1 | 1 | 0,1 | X | | | | | | |

2. Включение старшей тетрады в процесс обработки ТОС, что расширяет его разрядность в 1,5 раза. Особенность этого случая состоит в том, что при этом разрядность может превысить 16, 64, а это определяет необходимость в построении следующего яруса схем параллельного переноса ТОС.

# Сравнение реализаций

1. В случае использования условного переноса дополнительная задержка определяется операцией выбора нужного результата из трёх заготовленных и равна $t_{доп} = t_{ком}$=3τ, где $t_{ком}$ – задержка коммутатора, τ – задержка вентиля.

В случае включения старшей тетрады в процесс обработки ТОС возможны два варианта: увеличенная разрядность не превысит / превысит значение 16 или 64. В первом варианте дополнительная задержка не возникнет, а во втором варианте составит $t_{доп}$ = 2τ.

2. Лучшее решение по быстродействию для обработки старшей тетрады заключается в использовании ТОС с расширенной в 1,5 раза разрядностью. Однако, как показал анализ, это решение требует больших дополнительных аппаратных затрат. Выбор варианта обработки старших разрядов лежит на проектировщике умножителя.

# Заключение

Достоинством предложенного метода построения умножителя является регулярность его структуры. Так, собрав 4 умножителя 8x8 в общую структуру, получим умножитель 16×16; 4 умножителя 16×16 дадут умножитель 32×32, и т.д. Недостатком метода является то, что допустимые разрядности сомножителей ограничены рядом значений $3{\cdot}2^k$, $4{\cdot}2^k$, $5{\cdot}2^k$, где $k$ =1, 2... .

Сформулирована и доказана теорема о трёх рядах кодов, в которые выстраиваются результаты обработки подромбов; эти ряды могут быть далее обработаны трёхоперандным сумматором. Количество ярусов $r$ пирамиды ТОС равно глубине $k$ рекурсии при декомпозиции исходного ромба БЧП: $r=k$.

Для обработки старших разрядов предложены 2 варианта решения:

1) использование модифицированного условного переноса на основе вырожденных сумматоров;

2) расширение разрядности ТОС в 1,5 раза за счёт включении старших разрядов в общую разрядность ТОС. Наилучшим в смысле минимума задержки является последний вариант, однако для его реализации нужны большие аппаратные затраты.

Для схемотехнической реализации умножителя могут быть использованы программируемые логические матрицы (ПЛМ или ПМЛ), а также матричные кристаллы (БМК). ПЛМ (ПМЛ) проигрывают по быстродействию БМК, но у последних – проблемы с нагрузочной способностью элементов, а это в значительной мере ослабляет преимущество БМК.

# Литература

1. Паулин О.Н К построению быстродействующих арифметических устройств / О.Н. Паулин // Сб. Искусственный интеллект. – 2002. – № 3. – С. 314-322.
2. Паулин О.Н. Построение быстродействующих устройств умножения на основе многооперандных сумматоров / О.Н. Паулин, Н.И. Синегуб, А.М. Ляховецкий // Автоматика-2000 : праці міжнародної конференції з управління. – Львів, 2000. – С. 167-173
3. Паулин О.Н. К разработке умножителя на основе сумматора типа ромб / О.Н. Паулин // Искусственный интеллект. – 2006. – № 4.– С. 35-41.
4. Паулин О.Н. О свёртке трехрядных кодов / Паулин О.Н. // Управляющие системы и машины. – 2005. – № 5. – С. 68-72.
5. Паулин О.Н. К построению прикладной теории симметрических булевых функций / О.Н. Паулин // Искусственный интеллект. – 2005. – № 4. – С. 245-255.
6. Угрюмов Е.П. Цифровая схемотехника : учеб. пособие для вузов / Угрюмов Е.П. – [2-е изд., перераб. и доп.] – СПб : БХВ-Петербург, 2004. – 800 с.: ил.

# Literatura

1. Paulin O.N. Sb. “Iskustvennuy intellekt-2002”, specvyp. Doneck: IPII. “Nauka i osvita”. 2002. T.3. S. 314-322.
2. Paulin O.N.Praci mijnarodnoyi konferenciy z upravlinnya “Avtomatyka-2000”. Lviv. 2000. sekciya 7. S. 167-173.
3. Paulin O.N. Iskustvennuy intellekt. №4. 2006. Doneck: IPII. “Nauka i osvita”. 2006. S. 35-41.
4. Paulin O.N. Upravlyayuschie sistemy i mashiny. № 5. 2005. S. 68-72.
5. Paulin O.N. Iskustvennuy intellekt. № 4. 2005. Doneck: IPII. “Nauka i osvita”. 2005. S. 245-255.
6. Ugryumov E.P. Cyfrovaya sphemotehnika: Ucheb. posobie dlya vuzov. SPb: BHV-Peterburg, 2004. 800 s.

**_RESUME_**

***O.N. Paulin***

# *Features of Codes Compressing by Designing of Multipliers on the Base of the Rhombus Method*

The article is concerned with the pressing problem of decrease of multipliers delay. In the article formulation of the rhombus method is offered, which is described in the article of the same author "On Designing the Multiplier Based on the Rhombus Type Adder", in the journal “Artificial Intelligence”, 4’2006. In the method a rhombus of partial product bits is divided into subrhombuses of twice less dimension bits, and further they are operated on rhombus-like adders. In the present article by the example of 8×8 multiplier the features of compressing of ms bits of summable series of codes, which are the result of compressing of four 4×4 dimension subrhombuses of bits, are considered.

A theorem that at recursive decomposition of the results of compressing of four subrhombuses of bits of the previous processing stage (initially that is a rhombus of partial product bits) three rows of the codes are formed, is formulated and proved, that foundes the usage of a three-operand adder in the structure of $n \times n$ multiplier.

The following two options of processing of *n/2* ms digits of the results of subrhombuses of bits compressing are considered: 1) with the use of devolution adders with provision for two bits of unpositional carry from three-operand adder; 2) by including ms digits in the total capacity of three-operand adder. Minimal delay criterion analysis of the above-mentioned variants is performed, and usage recommendations are given.

With these means the method acquires finished character and allows to design multipliers on the basis of rhombus-like adders with the digit length of multipliers $n = m \cdot 2^k$, where $m$=3, 4, 5, and $k$ is level of recursion, $k$= 1,2, ... .

The advantages of the method is regularity of the multiplier structure, and also usage of symmetric function apparatus.

*Статья поступила в редакцию 31.05.2012.*